

## תרגיל בית 2

### תוכנית שרת לקוח מעל שרת לא אמין (UDP)

#### לצורך החזרת זמן בתצורות שונות

#### תאריך הגשה: 24.12.2011 למייל

#### המשימה

- עליכם להרחיב את אפליקצית שרת הזמן בכדי שתתמוך בשירותים הבאים:
- הלקוח יכול לשלוח 6 סוגי בקשות שונות, בהתאם לקלט המשתמש. להלן סוגי הבקשות:
    - a. GetTime – החזר את הזמן בתצורה של שנה, חודש, יום, שעה, דקה ושנייה (כמו בדוגמא המקורית).
    - b. GetTimeWithoutYear – החזר את הזמן בתצורה של שעה, דקה ושנייה.
    - c. GetTimeSinceEpoch – החזר את הזמן בתצורה של שניות החל מ-1.1.1970.
    - d. GetClientToServerDelayEstimation – החזר אומדן של ההשהיה בין הלקוח לשרת (ראו הנחיות להלן).
    - e. MeasureRTT – מדידת RoundTripTime (RTT) (ראו הנחיות להלן).
    - f. GetDayAndMonth – החזר את החודש והיום.
    - g. GetYear – החזר את השנה בלבד.
  - על השרת לעבד בקשות אלו ולהחזיר תשובה מתאימה, לפי סוג השירות המבוקש.
  - על השרת לרוץ בלולאה אינסופית ולחכות לקבלת בקשות מהלקוחות.
  - בדומה לאפליקצית שרת הזמן, הסוקטים יהיו Blocking ואין צורך לממש את האפליקציה כ-Non-Blocking.
  - לסעיפים ד ו-ה בלבד יש להשתמש בפונקציה **GetTickCount()**<sup>1</sup> במידה ומפתחים בסביבת windows (במקום פונקציית **time()**).
  - במידה ומשתמשים בסביבת Linux יש להשתמש בפונקציות המפורטות בקטע הקוד להלן. מבנה התשובה מוגדר בעזרת ה-**printf()**. הקוד המפרט את הפונקציות ב-Linux הוא:

```
• #include <stdlib.h>
• #include <sys/time.h>
• main()
• {
•     struct timeval tv;
•     struct timezone tz;
•     struct tm *tm;
•     gettimeofday(&tv, &tz);
•     tm=localtime(&tv.tv_sec);
•     printf(" %d:%02d:%02d %d \n", tm->tm_hour, tm->tm_min,
•           m->tm_sec, tv.tv_usec);
•     exit(0);
• }
```

<sup>1</sup>המוגדרת ב- <http://msdn.microsoft.com/en-us/library/ms724408%28VS.85%29.aspx>. שימו לב כי יש צורך ב:

- במידה ולא ממשים ב-LINUX ולא ב-WINDOWS, השתמשו בפונקציות "הרגילות" אולם שימו לב כי התוצאות שמתקבלות עבור סעיפים אלה (אם אינכם משתמשים בפונקציות אלה) אינן מהימנות עקב העובדה שהפונקציות עובדות ברזולוציה של שניות שלמות בלבד (ללא מילי-שניות לדוגמה).

### הנחיות לאומדן ההשהיה בין הלקוח לשרת (d)

בסעיף זה אנו מעוניינים למדוד את אחת ההשהיות אותן למדנו בשיעור הראשון.

- השתמשו בלקוח ה-TimeServer כך שישלח 100 בקשות "מה הזמן" (מותאמות לסעיף זה) רצופות. ולאחר מכן יקלוט 100 תגובות מהשרת.
- התשובה לבקשת "מה הזמן" בסעיף זה צריכה להשתמש ולהחזיר את הערך שמוחזר ע"י פונקציית `GetTickCount()` (כפי שהוסבר לעיל).
- הלקוח יעשה מיצוע (יחשב ממוצע) של הפרשי חותמות הזמן, כפי שנמדדו בשרת.

### הנחיות למדידת RTT (e)

- השתמשו בלקוח ה-TimeServer כך שישלח בסה"כ 100 בקשות "מה הזמן?" (לא רצופות!) ויקלוט 100 תגובות מהשרת. כלומר, בקשה, תשובה, בקשה, תשובה ובסה"כ 100 זוגות כאלו.
- עבור כל אחת מ-100 הבקשות, הלקוח יחשב את הזמן שחלף מרגע שליחת הבקשה ועד מועד קבלת התשובה המתאימה וימצע זמנים אלו עבור כל 100 ההודעות שנשלחו.

### שאלות תיאורטיות [10 נק']

- הריצו את התרגיל ובקשו בקשות `GetClientToServerDelayEstimation` רבות. האם ייתכן שהלקוח או השרת "ייתקעו"? אם לא, נמקו. אם כן, נמקו והציעו תיקון בתוכנית שלכם בכדי שהלקוח והשרת לא "ייתקעו".
- מבחינת ההשהיות השונות שאנו כבר מכירים (אותן 4 השהיות שנלמדו בכיתה), מה מבטא הגודל הממוצע אותו הלקוח מחשב על סמך תשובות השרת בבקשת `GetClientToServerDelayEstimation`? (המלצה: נתחו גודל זה מבחינה תאורטית ולא דווקא על סמך התוצאות המתקבלות מההרצה בפועל שעלולות להטעות, בפרט כאשר מבוצעות על אותו המחשב)

### מסמך נלווה [15 נק']

- יש להגיש בנוסף לקוד התכנית (קבצי ה-cpp) מסמך נלווה המתאר כיצד בחרתם לממש את האפליקציה, כלומר, סוגי ההודעות השונות, המבנה שלהן והפעולות השונות הנובעות מהן (זהו למעשה הפרוטוקול). בנוסף יש לצרף תצלומי מסך (`print screen`) של תוכנת ה-wireshark אשר יראו את סוגי ההודעות השונות וההתאמה שלהן למבנה שתיארתם באופן מילולי.
- כמובן שניתן גם לצרף את התשובות לשאלות לעיל לאותו מסמך.

### ההגשה היא ביחידים בלבד. בהצלחה!

- חובה להגיש תרגיל זה (אי הגשת התרגיל "תזכרה" בציון 0). אין לעשות תרגיל זה בקבוצה (זוג מוגדר כקבוצה).
- הגשה באיחור גוררת הורדת ציון של 15% לאחור של עד שלושה ימים, 30% עד שישה ימים וכך הלאה.